

REMARKS

Claims 1-29 are currently pending. Claims 1, 16, and 26 are independent. Claims 1, 3, 13, 16, and 26 are amended. Reconsideration of the action mailed November 13, 2006, is requested in light of the foregoing amendments and the following remarks.

The examiner rejected claims 1-2 and 5-6 under 35 U.S.C. § 102(b) as allegedly anticipated by U.S. Patent No. 4,924,408 ("Highland"). The examiner rejected claims 16-18, 20, and 25 under 35 U.S.C. § 102(b) as allegedly anticipated by U.S. Patent No. 5,423,041 ("Burke"). The examiner rejected claims 3-4 under 35 U.S.C. § 103(a) as allegedly unpatentable over Highland in view of Luke, "A Rule-Based Specification System for Computational Fluid Dynamics ("Luke"). The examiner rejected claims 26-28 under 35 U.S.C. § 103(a) as allegedly unpatentable over Highland. The examiner rejected claims 7-15 and 29 under 35 U.S.C. § 103(a) as allegedly unpatentable over Highland and Burke. The examiner rejected claims 19 and 21-24 under 35 U.S.C. § 103(a) as allegedly unpatentable over Burke.

Section 102 Rejections

Claim 1 stands rejected over Highland. Claim 1, as amended, recites a method that includes receiving a rule set as a single package in a processing system. A dependency graph is generated for the rule set. The dependency graph includes a plurality of ranked nodes. The nodes include entity nodes, attribute nodes, condition nodes, and rule nodes. A sequence of processing logic is generated for optimal processing of inputted facts according to a rank order of the nodes in the dependency graph.

The examiner states that Highland discloses the claimed dependency graph at FIG. 2a-2b and col. 5, lines 50-55. The applicant disagrees.

Highland discloses a method of converting a knowledge base and an inference engine into program source code. *See* col. 1, lines 18-23. The cited portions of Highland disclose translating rules of the knowledge base into a rule tree network.

Specifically, col. 5, lines 50-55 simply provide the brief description of the drawings for FIG. 2, which reads as follows:

FIG. 2 (comprised of FIG. 2a, FIG. 2b and FIG. 2c) illustrates how rules of a knowledge base (FIG. 2a) can be translated into a rule tree network (FIG. 2b), and how this rule tree network can be converted into compilable, procedural, program code according to the present invention (FIG. 2c).

FIG. 2a shows a rule of a knowledge base, specifically, "If $A > 10$ and $B < 0$ then display A". FIG. 2b shows a rule tree representing the rule of FIG. 2a. Finally, FIG. 2c, shows the generated code corresponding to the rule. *See* FIGS. 2a-2c; col. 5, lines 63-28. The rule tree represents one or more logically partitioned rules. *See* col. 6, lines 28-31. Thus, each node in the rule tree is a rule component. The rule tree is traversed beginning with the root node and labeled with a sequential index value, uniquely identifying each node in the rule tree. *See* col. 6, lines 37-42. A procedure is generated for the rule whereby the rule is evaluated according to the index values. *See* col. 44-52.

The cited txt and figures of Highland do not disclose or suggest the claimed dependency graph. The rule tree in Highland only includes rule components as nodes. The cited portions of Highland do not disclose or suggest that a dependency graph that includes a plurality of ranked nodes where the nodes include entity nodes, attribute nodes, condition nodes, and rule nodes, as required by claim 1.

The applicant submits that claim 1, as well as claims 2-15, which depend from claim 1, are in condition for allowance.

Claim 16 stands rejected over Burke. Claim 16, as amended, recites a method for automating business processes. The method includes receiving a rule set as a single package in a computer system. Logical conflicts within the rule set are determined. A logical conflict exists when two or more rules receiving the same inputs result in contradictory actions. The logical conflicts are resolved. A sequence of processing logic from the rule set is generated for optimal processing of inputted facts.

The examiner states that Burke discloses determining and resolving logical conflicts at col. 1, lines 59-64. The applicant disagrees. Col. 1, lines 59-64, read, in pertinent part, as follows:

The links represent the way in which facts and the results of tests are combined into rule conditions, the references made to these tests and rule conditions by other rules in the knowledge base, and the actions to be performed when a rule is "true".

The cited portion of Burke is found within the context of a background description of knowledge based systems as collections of data structures. *See* col. 1, lines 51-53. The data structures include nodes representing rules (and tests within the rules) and links between the nodes. *See* col. 1, lines 53-56. The cited portion of Burke discloses a way in which the links and nodes interact in order to process facts according to the rules. In particular, if the facts applied to a particular rule result in the rule being true (*i.e.*, the rule is satisfied), then a particular action is performed according to the rule. Likewise, if the rule is false under the tested facts (*i.e.*, the rule is not satisfied), a different action is performed.

Thus, the cited portion of Burke refers to evaluating a rule with respect to facts in order to proceed to a next operation (*e.g.*, a next node in the data structure). The cited portion does not disclose or suggest determining logical conflicts within a rule set where the logical conflicts exist when two or more rules receiving the same inputs result in contradictory actions. Whether a rule is true or false when evaluated does not disclose or suggest conflicts between distinct evaluated rules. Furthermore, the cited portion does not disclose or suggest resolving logical conflicts between rules once a logical conflict is determined.

The applicant submits that claim 16, as well as claims 17-25, which depend from claim 16, are in condition for allowance.

Section 103 Rejections

Claim 26 stands rejected as unpatentable over Highland. Claim 26 recites a computer program product that includes limitations corresponding to those of claim 1. The examiner rejects claim 26 using the same cited portions of Highland as claim 1. The examiner bases the obviousness rejection in that it is obvious to store computer readable code in a computer readable storage medium. The applicant notes, however, that claim 26 does not recite storing code in a medium. For the reasons set forth above with respect to claim 1, claim 26 as well as claims 27-29, which depend from claim 26, are in condition for allowance.

Conclusion

The applicant requests that all pending claims be allowed.

By responding in the foregoing remarks only to particular positions taken by the examiner, the applicant does not acquiesce with other positions that have not been explicitly addressed. In addition, the applicant's arguments for the patentability of a claim should not be understood as implying that no other reasons for the patentability of that claim exist.

Please apply the required one-month extension fee of \$60 and any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: March 12, 2007

/Brian J. Gustafson/

Brian J. Gustafson

Reg. No. 52,978

Customer No. 26161

Fish & Richardson P.C.

Telephone: (650) 839-5070

Facsimile: (650) 839-5071